

## CONCURRENT ENGINEERING WITH CONSTRAINT NETWORKS

Steve Minnaar<sup>1</sup> and Robert Reinecke<sup>2</sup>

### ABSTRACT

*Concurrent engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. It allows the consideration of all elements of the product's life cycle, from conception through disposal, including quality, cost, manufacturing, and customer requirements. Constraint networks as an approach to concurrent engineering provides considerable advantages over the conventional approaches to improve engineering. Constraint networks are difficult to implement though and function best when applied to a small design team concerned with designing within a family of related products or processes.*

---

1 Ph.D. student in Industrial Engineering, University of Stellenbosch, South Africa

2 Professor and Chairman of the Department of Industrial Engineering, University of Stellenbosch, South Africa.

## INTRODUCTION

Modern manufacturing faces severe national and international competition because of the growing number of innovative and skilled competitors in an already saturated market. The changing needs of the market call for an increase in the variety of product types. This implies design for smaller batch sizes and shorter delivery periods but with ever increasing emphasis on quality.

Early inventors designed, built and sold their products. But, as industry developed, work became specialized to the point where the designer was ill informed as to how his design will be manufactured, tested or serviced. Therefore design is today breadboarded, prototyped and built in pre-production runs before industrial production takes place. Between each stage redesign takes place routinely to also optimize the design by trial and error for all the characteristics other than pure technical feasibility. The process extends virtually throughout the life cycle of the product. This affects both the quality of the product in not conforming to the customers total requirements, and the productivity of effort by failing to achieve a fairly sound design first time.

Concurrent engineering, with constraint networks, can address this problem. Concurrent engineering aims at optimizing product development by the earliest possible integration of the company's knowledge, resources and experience regarding design, manufacturing, testing, quality, sales and product support. With proper and timely consideration of all necessary information, concurrent engineering reduces the number of design iterations, leading to improved time to market for new products. Constraint networks is an approach to concurrent engineering that facilitates the consideration of all necessary information, through the formulation, presentation and interactive satisfaction of product and process constraints, requirements and capabilities. [Minnaar, 92]

This paper first discusses engineering design and its difficulties. It then introduces concurrent engineering and the present approaches used. Finally the constraint network approach to concurrent engineering is discussed.

## ENGINEERING DESIGN

There are many schools of engineering which adequately describe the engineering process. This section is not an attempt to redefine the engineering process, or to refute systems engineering, value engineering or any of the other - *engineerings*. It discusses engineering design's problems as a background to the topic of concurrent engineering. According to Evans (87) there are three primary difficulties with design;

- \* the design process itself,
- \* the volume and variety of life-cycle information and
- \* the separation of life-cycle information.

### The Design Process

The design problem is basically separated into three stages; conceptual design, detailed design and analysis; and evaluation. Errors made at the conceptual phase of design propagate throughout the remainder of the design cycle, becoming more entrenched in the design as the cycle progresses. A small error at the conceptual level can snowball to a serious, hard to eradicate, problem in later stages. Each of the design stages are also further divided into subproblems. Each subproblem is subsequently solved until the overall design is acceptable.

This leads to sub-optimization, which could be detrimental to overall optimization, because global life-cycle information is not always considered.

### **Volume and Variety of life-cycle information**

The volume and variety of life-cycle information regarding a product is substantial. Life-cycle information includes the product specification, customer requirements, quality standards, military standards, manufacturing processes and constraints, purchasing requirements, testing information, development time-scales; and a whole range of other specifications, requirements, constraints and capabilities. Furthermore the information can be qualitative and quantitative and sometimes very difficult to dictate. Because of this volume and variety of information, designers concentrate on narrow functional domains and ignore the rest of the information (or at least postpone the consideration of some information until much later in the design process).

### **Separation of life-cycle information**

The different life-cycle functions (such as design, process planning, manufacturing, testing and servicing), are usually separated because they are the responsibilities of different departments in an organization. Interdepartmental communication is not always sufficient, leading to information loss and misunderstandings. Design changes are seldomly communicated back to the original designer, or other team members who should be aware of it. Each department designs with its own function as the first priority. Employees from the various departments become functional representatives, instead of being active team members of the project. This contributes to insufficient control of the development process and all the information involved.

One of the most adverse results of these three primary difficulties is that designers often defer considering life-cycle information until late in the design process, causing many iterations of the design process before the overall design is acceptable, as well as suboptimization of the product's function, quality and cost.

## **CONCURRENT ENGINEERING**

In 1988 the DARPA Initiative in Concurrent Engineering (DICE) was launched in the United States to encourage and research the practice of concurrent engineering in the US military and industrial base. By the end of 1991 DARPA (Defense Advanced Research Projects Agency) had spent US \$60 million on the DICE project. A consortium of more than a dozen industries, software companies and universities still participate in DICE. It is therefore not merely a new buzzword, but a renewed effort to improve engineering development.

Concurrent engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. It allows for the timely consideration of all elements of the product's life cycle, from conception through disposal. The term concurrent refers to the fact that traditional development phases should be run concurrently, so as to reduce the product's time to market. Concurrent development means that one should proceed on the assumption that a specified performance will be obtained from a component, even before that performance has been demonstrated. Such assumptions can only be made in an environment where there is certainty and availability of information. For example; a prototype will be built before the design effort is completed. The building of the prototype will only be useful if realistic assumptions can be made as to the results of the design effort.

Concurrent engineering therefore facilitates the acquisition, manipulation, presentation and consideration of all pertinent development information, so as to enable development phases to run concurrently, reduce design iterations, reduce time to market and increase product quality. The intelligent use of information can only increase conformance to customer requirements.

An example of the concurrent engineering philosophy in a typical electronic manufacturer: A new design would, after a number of iterations get to the point where a prototype (experimental model) needs to be built. A special workshop in the R&D facility would build the model from specially sourced components. The designers would then tweak this prototype until it works and from that information eventually produce a production file for the manufacturing department. More often than not, the manufacturing department takes months to get to full production because of teething problems with the new product. With concurrent engineering it is different. While designing the new product, the designers are in constant communication with manufacturing, quality and test, purchasing and marketing. The product is designed to compliment the manufacturing processes (or the processes can be adjusted in advance for the new product), it provides the necessary test points, uses approved components sourced by the purchasing department and conform to marketing's requirements. The prototype is built on the shopfloor so as to give manufacturing experience with the new product, test its design for manufacturability and ensures that it is indeed feasible to produce the product. It also gives manufacturing the opportunity to provide feedback to the designers for possible improvements in their design. From thereon the product is refined further. All development phases run in parallel, so that by the time that the design team completes the last details, the manufacturing plan, test and quality plan and facility and the purchasing would be complete as well. This way the new product can proceed seamlessly from R&D to manufacture. Even after the design is completed, the R&D engineers would still be involved with the product's manufacture to ensure that learn as much as possible from the manufacturing process so that they would be able to accommodate it even better in the future.

Concurrent engineering is to development what Just In Time is to manufacturing. Development information is communicated just in time and in small batches. Instead of one department completing their function in the development process before handing it over to the next department, the department continually communicates their progress to the other involved departments. This reduces the time for feedback on design and facilitates the consideration of information from other development functions. It also prevents extensive redesign or rework because one department has based all their work on an incorrect premises (i.e. the total redesign of a PCB layout because the PCB size is incompatible with the automated assembly line).

The confusion between systems engineering and concurrent engineering necessitates some comment here. Systems engineering is an approach which views an entire system of components as an entity, rather than simply as an assembly of individual parts; i.e. a system in which each component is designed to fit in properly with the other components rather than function by itself. [Machol, 65] Concurrent engineering is not in opposition to systems engineering, it actually supports it. Concurrent engineering focuses on the concurrent development of product phases, with more emphasis on cradle to grave information being considered from the beginning. Where systems engineering concentrates the design of systems, concurrent engineering applies to any product development, from its conception through to its disposal. Concurrent engineering emphasizes the feedback of information from later stages in the product's life for improvement of new designs. It is therefore a more complete, flexible and

modern approach to engineering that also supports system engineering; if the latter's principles apply to the product in question.

## **CURRENT APPROACHES TO IMPROVE DESIGN ENGINEERING**

There are many focused approaches to improve design engineering where the specific information of some single facet of the product's life-cycle is incorporated into the design phase. Examples of these are Design for Assembly (Manual, Automatic and Robots), [Boothroyd, 83 and 84; Warnecke, 88; Poli, 86; Schuch, 89], Design for Manufacture [Donovan, 89; Swift, 89], Design for Production [Suh, 88], Design for Automated Manufacture and Assembly [Scarr, 86], Design for Testing [Gabay, 90] and Process Planning [Liau, 91].

Even these focused approaches have lead to significant improvements in the design process. Boothroyd [83a] claims a 20% to 40% drop in manufacturing cost and an assembly productivity rise of 100% to 200% with the utilization of Design for Assembly. These figures are very plausible considering that upwards of 70% of a product's manufacturing costs is determined during the design phase. [O'Grady, 90]

The current approaches to improved engineering are however not able to totally overcome the difficulties mentioned earlier. The main reasons for their lack of complete success are:

### **Design Teams**

Design teams is the traditional approach to improved product development. Although design teams have numerous disadvantages, it will always be the starting point for any development project, because humans are inherently involved. What is important though is that it must be realized that setting up a team does not solve all engineering problems. Design teams need tools and techniques to optimize their combined effort. It is also not satisfactory to just have teams to review individual efforts of team members, because this will merely lead to numerous design iterations.

### **Textbooks, Check Lists and Procedures**

The most widely used approach to design is to consult textbooks, quality specifications, check lists or compiled design procedures. The lack of flexibility and the bulk of printed matter make effective consulting difficult and ineffective. Many organizations have their capabilities and requirements compiled into design manuals, but because of the sheer size, magnitude and diversity of information, it is hardly ever used. Updating and validating information on printed matter is time consuming and costly.

The amount of useful information at a specific point in time is usually a very small percentage of all the information presented. The designer still has to negotiate all the information in order to reach and use the small percentage applicable to his situation. Very often there are conflicting information from the various textbooks or manuals.

Another problem with checklists and procedures is that they are post-design approaches. Only after the design has been completed would the designer be made aware of possible problem areas. This inherently leads to a trial and error approach. Boothroyd and Dewhurst's approach rely upon taking apart an already manufactured product and improving it. [Boothroyd, 83 and 84] Although they have much success in improving existing designs, it is of little help to the first off design.

### Process Planning and Manufacturing Simulation

These approaches are useful for process planning, but they provide little (if any) feedback to the designer as to how he can improve his design. Again it a post-design approach that inherently leads to trial and error.

### Knowledge Based Expert Systems

Many researchers believe expert systems to be the answer to concurrent engineering problems. [Maher, 84; Wallbank, 87] Although expert systems can be very useful, it has a number of inherent disadvantages for true design purposes. Expert systems are rulebased, making it difficult to represent the relationships between all the life-cycle constraints on a product. It requires the designer to start his design at a specific point and then guides him through the rules and expertise to some predefined solution. This approach is not acceptable for innovative, original design. There is a need for the expert's knowledge, but not necessarily his path of reason or solution.

The acquisition of the knowledge for the knowledge base and the induction of rules can also be very time consuming and difficult. [Van Terheyden, 87]

It can therefore be seen that current approaches to improved engineering have inherent disadvantages. The complexity of the design processes furthermore increases significantly with each characteristic of the design that must be optimized simultaneously. Totally concurrent design is hugely complex. It is thought that large scale concurrent design truly optimizing all individual requirements will call for computer power and capacity not yet available.

### CONSTRAINT NETWORKS

Constraint networks is the approach to concurrent engineering proposed here. Constraint networks are particularly useful when the product being designed is part of a larger family of products, or where the organization use the same manufacturing processes for all products. It has several advantages over the traditional approaches mentioned earlier.

Design can be seen as the process to satisfy a set of constraints. [Ohsuga, 89; Balleo, 85] Design through constraint satisfaction has been applied to, or proposed for, a wide variety of design domains, including: electrical circuits [Sussman, 80], mechanical design [Kimura, 87], structural design of buildings [Maher, 84] and finite element analysis [Chang, 88].

It is useful to have a formal way to represent models, i.e. in the form of a directed network. This is exactly what constraint networks achieve. A constraint expresses what is allowed, possible or required of the parameters. The constraint network is a collection of all the constraints which are interconnected because of the relationships that exists between the constraints. [Mackworth, 87]

There are two approaches to constraint networks; procedural and declarative. In the procedural approach, a decision is made *a priori* about the order in which the parameters involved in a constraint acquire their value. This approach limits the creative ability of the designer, because there is a rigid distinction between input and output parameters. The second and preferred approach is the declarative approach. In this approach no commitment is made about the order in which the parameters are evaluated. A declarative language requires only the statement of relevant constraint relationships, and the computational organization is

performed more or less automatically. [Sussman, 80] The declarative approach will be used for constraint networks.

There are three basic steps in the utilization of constraint networks for design purposes. They are constraint formulation, constraint propagation and interactive constraint satisfaction.

### **Constraint Formulation**

The generation of the constraints is usually one of the most difficult aspects of utilizing constraint networks. An incorrect or insufficient constraint network greatly reduces the chances of an optimized design. The induction of constraints should therefore be approached in a most responsible way. [Van Terheyden, 87] Constraints can be classified as either basic (equations) or complex (conditional constraints, constraint generators and nested constraints). [Zima, 86]

### **Constraint Propagation**

Propagation is the generation of new constraints from already formulated constraints. Because of the interrelationships between constraints in the network, the value of one parameter may influence the values of others. Changes in parameter values may propagate throughout the network. This ability of propagation makes constraint networks unique and enables the constraint network to support non-directional inference. The user of the constraint network can therefore see the changes made in linked constraints (parameters) as the initial change is made. [O'Grady, 90]

### **Interactive Constraint Satisfaction**

In interactive constraint satisfaction the user assigns the parameters their values (in any sequence), while a computerized interpreter tests the assignment with all existing assignments. The assignment of values and testing for consistency can be allocated in any combination to humans and computers, but the combination as just mentioned is the most desirable for optimized design. It is best for the user to retain control over the design process, while the computer does the menial, repetitive and laborious work. The alternative is automatic design, but this has numerous disadvantages, including the tendency to become np-complete as the number of parameters increase. This is an interactive process, because the user is utilized *a priori* as part of the inference mechanism.

### **Constraint based languages**

There are at present a number of constraint based languages for design with constraint networks. Examples are Sketchpad, ThingLab, IDEAL, CONSTRAINTS and TK!Solver. [O'Grady, 90] They all share the disadvantage that they were developed for narrow application areas.

There are a number of principal requirements for a computerized system to support concurrent engineering [Young, 91]:

- \* it should be flexible enough to allow the design problem to be approached from a variety of viewpoints;
- \* it should allow the designer to design in the absence of complete information;
- \* it should handle the large volume, variety and interdependence of life-cycle information;
- \* it should readily interface to database management and CAD systems;
- \* it should have a good user interface and be able to explain itself in a manner comprehensible to humans;

- \* it should support design audits.

Considering the above requirements, North Carolina State University developed GALILEO [O'Grady, 90] in Prolog with the following added advantages over the other systems; domain independence, the ability to handle a wide variety of object and constraint types, and the ability to explain its reasoning to the user. A more powerful system, SATURN, has also been developed since for bigger applications. [O'Grady, 92] Minnaar did the debugging of SATURN, wrote the reference and user's manual for SATURN, and developed a practical demonstration while visiting NC State in 1992 for his M.Eng thesis.

### **Practical Implications of the Constraint Network**

The constraint network does not automatically design the product. This would be computationally impractical for complex concurrent engineering design. The designer does all the design work, while via constraint monitoring, the constraint network ensures that the designer is aware of all violated constraints, as well as possible ways to correct the violated constraint. The objective when using a constraint network is to find a set of variables that does not violate any of the constraints. A well designed constraint network helps the designer because it indicates the specific constraint violated and suggests ways to satisfy the given constraint.

With constraint networks the designer iteratively assigns values to the parameters until global consistency of the constraint network is achieved (if there exists a feasible solution). Because of constraint propagation the designer would not have to assign values to all the parameters. The intelligent constraint network would propagate correct values to some of the parameters because of the interrelationships between the constraints. Constraint networks therefore not only aids in achieving an optimum design that considers all life-cycle information in less time than conventional systems, it also reduces the burden on the designer.

Because of the complexities involved in formulating such a constraint network, the ideal application would be for designs within a family of related designs. Examples are variants of printed wiring boards, railway bridge design, paper mill designs and boiler design. O'Grady [90] gives a simple, but thoroughly explained, example of a constraint network that expresses the approach of Boothroyd and Dewhurst for Design for Assembly. [Boothroyd, 83b] The constraint network has these added advantages over Boothroyd's approach in that:

- \* it gives advice on how to improve the design,
- \* the designer can interact with the system to improve the design and
- \* the designer can approach the problem from any viewpoint without the need to follow a fixed procedure.

### **CONSTRAINT NETWORKS AND INDUSTRY**

Constraint networks as an approach to concurrent engineering is very easy to justify academically. Applying the technology to industry is more difficult.

Plessey-Tellumat, a local electronics manufacturer, has been implementing concurrent engineering for some months now and Minnaar has been advising them. There has been considerable progress, as well as cost and time savings. Plastamid, a plastics manufacturer will also be assisted by Minnaar with the implementation of concurrent engineering. The experience at these two manufacturers illustrates the applicability of constraint networks as a feasible approach to concurrent engineering.



Plessey-Tellumat is a medium sized organization that does a lot of high technology R&D. They have decentralized departments based on function and therefore really needs concurrent engineering because of the lacking communication and consideration between departments. They are however not suited for constraint based languages. A much simpler form of constraint network is needed that would not require as much extra input from the various disciplines. Because of the relative large number of design team members, few individuals feel responsible for the product and could therefore not care whether they consider other product information or not. The general attitude is that of functional representation; as long as they complete their part of the design to the satisfaction of their department, they are content. Few team members feel that they "own" the total end product.

The development of proper cross disciplinary team dynamics, a communication network and support system for design information has more priority than a constraint modeling environment at this stage. To convince an engineer in such an organization that he should code all his constraints into a constraint network that would be used by someone else is almost impossible. It is much simpler for them to use the telephone when they have problems. The disadvantage is of course that the same design mistakes are made time again and that designers do not consider each other as much as they should; they only use the telephone when they think that they need to consider someone else. Experience has proved this to be a fatal approach. Plessey-Tellumat's first steps towards concurrent engineering is to build efficient design teams and to develop a complete electronic communication system between the various team members. Once this is established, they can start using constraint network like approaches that can be on the electronic communication system.

A small organization with only a handful of design engineers is more suitable for the immediate implementation of constraint networks. Individual designers are usually responsible for a wider range of design activities and therefore experience the problems of insufficient or badly presented information personally. There the designers can set up networks to help themselves at first and later develop it further to include other design team members. Because design teams are smaller, they can see the relation between the customer's requirements, the product's specification, their work, the final product and the financial implications of their efforts. It is therefore easier to visualize the advantages of a constraint network approach.

Although concurrent engineering is universally applicable, its constraint network approach is not. Careful analysis of the type, scope and size of an organization is necessary before one should attempt the constraint network approach.

## CONCLUSION

Successful engineering design proves to be very complex when many aspects of life-cycle information has to be taken into consideration. It is becoming evident that concurrent engineering promises to be effective in approaching total optimization of design. Because of the disadvantages of current approaches to concurrent engineering, better solutions has to be found. Constraint networks is a viable approach where there is a small design team with designers who are directly responsible for the whole product, and not just their own function.

The authors are indebted to the Foundation for Research Development, Plessey-Tellumat, AECI and the University of Stellenbosch for their financial support to make this research and development possible.

## REFERENCES

- [Balleo, 85] Balleo, G and Contro, R. Short Communication: Iterative design procedure in the presence of technological constraints, *Advances in Engineering Software*, Volume 7, Number 2, 1985, p 94
- [Boothroyd, 83a] Boothroyd, G and Dewhurst P. Design for Assembly: Selecting the Right Method, *Machine Design*, 10 November 1983, pp 94-98
- [Boothroyd, 83b] Boothroyd, G and Dewhurst P. Design for Assembly: Manual Assembly, *Machine Design*, 8 December 1983, pp 140-145
- [Boothroyd, 84a] Boothroyd, G and Dewhurst P. Design for Assembly: Automatic Assembly, *Machine Design*, 26 January 1984, pp 87-92
- [Boothroyd, 84b] Boothroyd, G and Dewhurst P. Design for Assembly: Robots, *Machine Design*, 23 February 1984, pp 72-76
- [Chang, 88] Chang, S.C. and Lin, T.W. Constraint relation implementation for finite element analysis from an element basis, *Advances in Engineering Software*, Volume 10, Number 4, 1988, pp 191
- [Donovan, 89] Donovan, J.R. *Training for design for manufacture*, Proceedings of the Institution for Mechanical Engineers, 1989, Volume 203, pp 47-53
- [Evans, 88] Evans, B. Simultaneous Engineering, *Mechanical Engineering*, 1988, Volume 110, No 2, New York: American Society of Mechanical Engineers
- [Gabay, 90] Gabay, J. How much can Design-for-Test reduce the need for testing? *Computer Design*, 1 September 1990, pp 94 - 112
- [Kimura, 87] Kimura, F, Suzuki, H, Ando, H, Sato, T and Kinosada, A. *Variational Geometry Based on Logical Constraints and its Applications to Product Modeling*, Annals of the CIRP, 1987, Volume 36, No 1, pp 65-68
- [Liau, 91] Liau, J and Young, R.E. *PACE: A Process Planning and Concurrent Engineering System for Printed Circuit Boards*, Progress Report PR-91-24, North Carolina State University, 1991
- [Machol, 65] Machol, R.E. (Ed), *System Engineering Handbook*, McGraw-Hill Book Company, New York, 1965
- [Mackworth, 87] Mackworth, A. Constraint Satisfaction, in S. Shapiro (ed.), *The Encyclopedia of Artificial Intelligence*, NY: Wiley, 1987, pp 205-211
- [Maher, 84] Maher, M.L., Sriram, D. and Fennes, S.J. Tools and Techniques for Knowledge Based Expert Systems for Engineering Design, *Advances in Engineering Software*, Volume 6, Number 4, 1984, pp 178 - 188

- [Minnaar, 92] Minnaar, S.J., *Application of AI in Manufacturing: Constraint Networks for Concurrent Engineering, applied to Surface Mount Technology*, Unpublished M.Eng Thesis, University of Stellenbosch, 1992
- [O'Grady, 90] O'Grady, P and Oh, J.S. *An Artificial Intelligence Constraint Network Approach to Design for Assembly*, Technical Report TR-90-14, North Carolina State University, 1990
- [O'Grady, 92] O'Grady, P and Young, R.E. *Constraint Nets for Life Cycle Engineering: Concurrent Engineering*, Proceedings of the National Science Foundation Grantees Conference, January 1992
- [Ohsuga, 89] Ohsuga, S. Toward intelligent CAD systems *Computer Aided Design*, June 1989, Volume 21, Number 5, pp 315 - 320
- [Poli, 86] Poli, C, Graves, R and Groppetti, R. Rating Products for Ease of Assembly, *Machine Design*, 21 August 1986, pp 79-84
- [Scarr, 86] Scarr, A.J. *Product Design for Automated Manufacture and Assembly*, Annals of the CIRP, 1986, Volume 35, No 1, pp 1-5
- [Schuch, 89] Schuch, L. DFA Promises and Delivers, *Assembly Engineering*, 1989, Volume 5, pp 18-21
- [Suh, 88] Suh, N.P. *Basic Concepts in Design for Producibility*, Keynote Paper, Annals of the CIRP, 1988, Volume 37, No 2, pp 559-567
- [Sussman, 80] Sussman, G and Steele, G. CONSTRAINTS - A Language for Expressing Almost-Hierarchical Descriptions, *Artificial Intelligence*, 1980, Volume 14, pp 1-39
- [Swift, 89] Swift, K.G. *Techniques in design for manufacture*, Proceedings of the Institution for Mechanical Engineers, 1989, Volume 203, pp 25-28
- [Van Terheyden, 87] Van Terheyden, A.G.R. and Chalcraft, D.A. Combining inductive and deductive reasoning, *Computer-Aided Engineering Journal*, February 1987, pp 24 - 28
- [Wallbank, 87] Wallbank, J and Whittingham, N.A. Artificial Intelligence within Austin Rover, *Computer-Aided Engineering Journal*, June 1987, pp 113 - 116
- [Warnecke, 88] Warnecke, H.J. *Design for Assembly - Part of the Design Process*, Annals of the CIRP, 1988, Volume 37, No 1, pp 1-4
- [Young, 91] Young, R.E., Greef, A and O'Grady, P. *An Artificial Intelligence-based constraint Network System for Concurrent Engineering*, Project Report PR-91-01, North Carolina State University, 1991
- [Zima, 86] Zima, H.A. A constraint language and its interpreter, *Computer Languages*, 1986, Volume 11, No 2, pp 65-83